VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Implementace IPv6 v počítačové síti Ostravské univerzity
IPv6 Implementation on University of Ostrava Computer Network

2010 Radim Hlávka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Radim Hlávka**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R059 Mobilní technologie

Téma: Implementace IPv6 v počítačové síti Ostravské univerzity
IPv6 Implementation on University of Ostrava Computer Network

Zásady pro vypracování:

Cílem práce je kompletní nasazení komunikace v počítačové síti Ostravské univerzity i na IPv6, zjištění dalších možností využití a dopracování dokumentace pro uživatele.

1. Předělat ipv6 směrování na centrální síťové prvky Cisco Catalyst 6509.
2. Implementovat překlad jmen v DNS na ipv6.
3. Implementovat stavové přidělování ipv6 adres (DHCP).
4. Nasadit ipv6 na produkčních serverech, kde to bude, po dohodě se správci, možné/vhodné.
5. Zabezpečení ipv6 sítě (centrální firewall, lokální zabezpečení na serverech/stanicích).
6. Prozkoumání možností IP mobility a hlasových služeb na ipv6 síti Ostravské univerzity.
7. Doplnění dokumentace k ipv6 na Ostravské univerzitě.
8. Nakonfigurování testovacího serveru pro ipv6 komunikaci (www,ftp .. etc.).

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Ivan Doležal**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky OSTRAVA

doc. Dr.Ing. Eduard Sojka
vedoucí katedry

prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

…..... …..…..

Děkuji Ing. Ivanu Doležalovi za odborné vedení při psaní této bakalářské práce a své rodině za podporu a trpělivost.

Abstrakt:

Předkládaná práce si klade tyto cíle: teoreticky seznámit čtenáře se základy nastupujícího standardu IPv6 (zejména s ohledem na jeho směrování); popsat získané zkušenosti s nasazením jeho stávajících implementací v prostředí produkční sítě Ostravské univerzity; poskytnout uživatelům a správcům základní síťové služby potřebné pro jeho provoz a správu; zabývat se otázkami jeho zabezpečení.

Klíčová slova:

IPv6, směrování, DHCPv6, DNS, Cisco FWSM, Linux

Abstract:

The presented thesis aims to introduce reader to basics of emerging IPv6 standard (with stress on its routing), to gain experience with the deployment of its current implementations in the production network at The University of Ostrava and to provide users and administrators with basic network services necessary for its operation and management; to deal with its security matters.

Keywords:

IPv6, routing, DHCPv6, DNS, Cisco FWSM, Linux

Seznam použitých zkratek:

AH	Authentication Header
ARP	Address Resolution Protocol
AS	Autonomní systém
ASDM	Adaptive Security Device Manager
BGP	Border Gateway Protocol
BGP4+	Border Gateway Protocol version 4+
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DNS	Domain Name System
DUID	DHCP Unique Identifier
EGP	Exterior Gateway Protocol
ESP	Encapsulation Security Payload
FWSM	Firewall Switch Module
IA	Identity Association
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
IETF	Internet Engeneering Task Force
IGP	Interior Gateway Protocol
IKEv2	Internet Key Exchange Protocol version 2
IOS	Internetwork Operating System
IP	Internet Protokol
IPS	Intrusion Prevention Systems
IPsec	Internet Protocol security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate System
ISC	Internet System Consortium
LSA	Link State Advertisement
MAC	Media Access Control
MX	Mail Exchange
NEMO	Network Mobility
NS	Name Server
OS	Operační systém
OSPF	Open Shortest Path First
OSPFv3	Open Shortest Path First version 3
OU	Ostravská univerzita

PTR	Pointer Record
QoS	Quality of Service
RFC	Request for Comments
RIPng	Routing Information Protocol – next generation
SEND	Secure Neighbor Discovery
SOA	Start of Authority
TCP	Transmision Control Protocol
TTL	Time to Live
VLAN	Virtual Local Area Netwok
VPN	Virtual Private network

Obsah

1. Úvod

Téma bakalářské práce jsem vybral s ohledem na svou profesi správce počítačové sítě. IPv6 jsme na naší síti již částečně provozovali, samozřejmě paralelně s protokolem IPv4, jeho praktické využití na počítačové síti ale bylo minimální. Svou prací jsem tedy chtěl situaci posunout směrem k většímu využití této technologie. Samotné fyzické nasazení protokolu do provozu ale nestačí. Předpokládám proto využití této práce i jako teoretický základ při uvádění do problematiky IPv6 technické pracovníky, starající se o uživatelské stanice, a případné zájemce z řad administrátorů serverů i koncových uživatelů.
Bakalářskou práci jsem rozdělil do sedmi kapitol. V kapitole 2 popisuji základní prvky IPv6 protokolu. Následující kapitola 3 popisuje možnosti směrování a směrovací protokoly. V kapitole 4 se zabývám principem fungování přidělování adres a implementací DHCPv6 v síti OU. V kapitole 5 je pak popsán systém překladu jmen a adres – DNS a jeho nasazení. Kapitolu 6 věnuji problematice zabezpečení IPv6 sítí a kapitolu 7 mobilitě.

2. O IPv6

Počátky vývoje *Internetového Protokolu verze 6* se datují v devadesátých létech 20. století. Motivace vývoje byla zřejmá – s prudkým rozvojem Internetu bylo jasné, že dřív nebo později IPv4 adresy dojdou. Cíle pro vývoj IPv6 stanovila IETF (vyvíjí a podporuje internetové standardy) takto:

- obrovský adresní prostor, který by vydržel „na věky“
- tři druhy adres:
    - unicast (individuální)
    - multicast (skupinové)
    - anycast (výběrové)
- jednotné adresní schéma pro Internet i vnitřní sítě
- hierarchické směrování ruku v ruce s hierarchickou adresací
- optimalizace pro vysokorychlostní směrování
- zvýšení bezpečnosti (mechanismy pro šifrování a autentizaci)
- automatická konfigurace
- podpora pro služby QoS
- podpora mobility

- plynulý a bezproblémový přechod z IPv4 na IPv6

Výsledky vývoje byly definovány koncem roku 1995 v RFC 1883: *Internet Protocol, Version 6 (IPv6) Specification* a jeho příbuzných. Nicméně od teorie k praxi – implementaci IPv6 do zařízení, uběhlo hodně vody. Důvodem bylo především zavedení beztřídního adresování u IPv4 (snížení potřeby IPv4 adres) a obecně chladnější přístup firem k této technologii. Vývoj a implementace přesto zvolna pokračovala, byly definovány další RFC, nicméně problém s volnými IPv4 adresami se opět vynořil. Například ve zpravodaji sdružení Cesnet *Datagram* (březen 2010) se uvádí, že zbývá méně než 10% volných IPv4 adres s pravděpodobným datem úplného vyčerpání - říjen 2012. Je velmi pravděpodobné, že toto datum není úplně přesné, ale skutečnosti se blížit bude. Vzhledem k tomu, že IETF nevyvíjí jinou alternativu k IPv4, je postupné rozšíření IPv6 nevyhnutelné (stát se ale může samozřejmě cokoliv a IPv6 klidně „umře“).

Shrnutí základních principů

Požadavek na obrovský adresní prostor v IPv6 vedl ke stanovení adresy na 128 bitů, což je čtyřnásobek délky IPv4 adresy. K dispozici je tak teoreticky $3{,}4*10^{38}$ adres, a to je opravdu hodně moc. Na identifikátor rozhraní připadá 64 bitů, takže v každé podsíti můžou být miliardy počítačů. Více o adresování v kapitole 2.2. Základní hlavička datagramu má v IPv6 konstantní délku. Volitelné položky se přesunuly do samostatných hlaviček, které se řetězí za základní hlavičku. Takto může směrovač co nejrychleji zpracovat informace určené pro něj a ostatními údaji se nezabývá. Více o formátu datagramu v následující kapitole 2.1. Požadavek na automatickou konfiguraci byl vyřešen nabídnutím dvou variant a to stavovou konfigurací (DHCPv6) a bezstavovou konfigurací (viz kapitola 4).Bezstavová konfigurace využívá i mechanismů *objevování sousedů*, které primárně slouží pro hledání fyzických adres sousedních počítačů (ekvivalent ARP v IPv4). Více v kapitole 2.4. K zajištění bezpečnosti přenosu datagramu slouží hlavička autentizační a hlavička šifrovací. Autentizační pomůže ověřit odesílatele datagramu, šifrovací přidává možnost zašifrovat obsah datagramu. Pro podporu mobility se v IPv6 využívají *domácí agenti*. To jsou směrovače z domácí sítě mobilního uzlu, které jej zastupují v době jeho jiného umístění. Podrobnější popis tohoto principu je v kapitole 7. Pro usnadnění koexistence IPv6 s IPv4 byly definovány některé nástroje jako je tunelování či překlad datagramů. Podrobný popis principů fungování IPv6 je popsán v [1] a [2].

2.1 Datagram

Má i ve verzi IPv6 tradiční formát – hlavičku a nesená data. Jak už jsem se zmínil výše, u verze IPv6 je ale **základní** hlavička minimální a s neměnnou délkou. Veškeré rozšiřující a doplňující údaje se přesunuly do rozšiřujících hlaviček.

Na následujícím obrázku je formát základní hlavičky:

Verze	Třída provozu	Značka toku	
Délka dat		Další hlavička	Max. skoků
Zdrojová adresa			
Cílová adresa			

Ilustrace 1: Hlavička IPv6 datagramu

Položka *Verze (Version)* obsahuje identifikátor protokolu, v tomto případě hodnotu 6. *Třída provozu (Traffic class)* definuje prioritu datagramu nebo jeho zařazení do určité přepravní třídy. V současné době se tato položka využívá při poskytování diferencovaných služeb (diffserv), kdy v závislosti na obsažené hodnotě může být datagram přednostně zpracován, či naopak jeho zpracování může být odloženo. Implicitní hodnotou je 0. Další položka Značka toku (Flow label) je novinkou v IPv6 a zatím není přesně definována. Obecně by ale takto měl být označen proud datagramů se společnými vlastnostmi (odesílatel, příjemce, požadavky na spojení). Směrovač by takto identifikoval datagramy ze stejného toku a naložil by s nimi stejně jako s předešlými a urychlí tímto rozhodovací proces. Údaj o počtu bajtů následujících za základní hlavičkou je obsažen v položce Délka *dat (Payload lenght)*. Tato položka je dvoubajtová, čili maximální délka datagramu (bráno bez základní hlavičky) může být 64 KB. Existuje ovšem výjimka v podobě rozšiřující hlavičky *Jumbo obsah*. *Další hlavička (Next header)* říká jaké rozšiřující hlavičky následují za základní hlavičkou, či jaká data za ní následují. Obdobou TTL ve světě IPv4 je položka Maximální počet skoků (Hop limit). Je zde uvedeno, kolikrát datagram může maximálně projít na své cestě přes nějaký směrovač. Směrovač při každém průchodu datagramu tuto hodnotu sníží o 1 a v případě vynulování pošle odesílateli zprávu o vypršení limitu. Základní hlavičku ukončuje dvojice polí *Zdrojová adresa (Source address)* a *Cílová adresa (Destination addres)*, které zabírají celých 80% z této hlavičky.

Zřetězení hlaviček

Na základní hlavičku můžou tedy navazovat hlavičky rozšiřující. Kód následující hlavičky (nebo typu nesených dat) je obsažen v poli *Další hlavička*. Každá z rozšiřujících hlaviček má svou položku *Další hlavička*, čímž jde libovolně hlavičky řetězit dále. V poslední z hlaviček datagramu je v tomto poli údaj o nesených datech. Nejdůležitější kódy jsou popsány v následujících tabulkách:

Kód	Popis
0	volby pro všechny (hop-by-hop options)
43	směrování (routing header)
44	fragmentace (fragment header)
50	šifrování obsahu (ESP, EncapSecurity Payload)
51	autentizace (AH, Authentication Header)
59	poslední hlavička (no next header)
60	volby pro cíl (destination options)
135	mobilita (mobility header)

Tabulka 1: Kódy rozšiřujících hlaviček

Kód	Typ nesených dat
6	TCP
8	EGP
9	IGP
17	UDP
46	RSVP
47	GRE
58	ICMP

Tabulka 2: Kódy typů nesených dat

Tato koncepce zřetězení je pružná a teoreticky by měla být i úsporná. Aby byla úsporná i v praxi, byl definován postup zřetězení jednotlivých rozšiřujících hlaviček. Tento postup pomůže zejména při průchodu datagramu směrovačem, který se tak nemusí prokousávat přes několik hlaviček než se dostane k potřebným údajům. Pořadí je definováno následně:

- základní hlavička
- volby pro všechny
- volby pro cíl (pro první cílovou adresu v hlavičce směrování)
- směrování
- fragmentace
- autentizace

- šifrování obsahu
- volby pro cíl (pro konečného příjemce datagramu)
- mobilita

Z tohoto pořadí je zřejmé, že nejdříve se připojí hlavičky zajímavé pro průchozí uzly na cestě datagramu a za nimi se pak připojí hlavičky určené koncovému příjemci.

2.2 Adresní prostor

Délka a podoba IPv6 adresy je popsána v dokumentu RFC 4291: *IP version 6 Addressing Architecture*. Definovány jsou tři typy adres:

- unicast – adresa rozhraní, které se data doručí přímo
- multicast – pomocí něj se adresují skupiny počítačů. Data odeslaná na tuto adresu se doručí všem členům skupiny.
- anycast – pomocí něj se také adresují skupiny počítačů, ale data se doručí jen jedinému členovi – tomu, který je odesílateli nejblíže.

Chybí, ze světa IPv4 známá, všesměrová broadcast adresa. Její funkci převzali skupinové adresy.

IPv6 adresy se přidělují jednotlivým rozhraním na zařízení pracující v síti a takovéto rozhraní může mít adres hned několik. Zařízení ve společné podsíti mají stejný prefix sítě. Délka adresy byla definována na 128 bitů. Zapisuje se jako osm skupin po čtyřech číslicích šestnáctkové soustavy navzájem oddělených dvojtečkami. Vypadat může takto:

2001:0718:1005:0606:0000:0000:0000:abcd

Vzhledem k délce adresy je povolena možnost zkracování. Lze vynechat 0 na začátku čtveřic, nebo lze vynechat celou 0000 čtveřici, nebo dokonce několik čtveřic vedle sebe a to zapsáním „::“ místo nich. Takže výše napsaná adresa vypadá po zkrácení takto:

2001:718:1005:606::abcd

To už je přece jen o něco kratší zápis, nicméně na zapamatování stále náročný.

Některé přechodové mechanismy mezi světy IPv4 a IPv6 potřebují namapovat IPv4 adresu. Způsob je takový, že prvních 80 bitů je nulových, následuje 16 bitů jedniček a v posledních 32 bitech je zapsaná v hexadecimální formě IPv4 adresa. Například má adresa 195.113.106.169 by namapovaná vypadala takto:

::ffff:c371:6aa9

Fungoval i starší způsob mapování, kdy na 96 nulových bitů navazovala 32 bitová přepsaná IPv4 adresa v původním formátu. Tento způsob byl ale v RFC 4291 prohlášen za neplatný.

Prefixy

Prefixy vyjadřují příslušnost rozhraní ke společné podsíti, všechna zařízení ho mají stejný. Obecně platí že poskytovatelé mají prefixy kratší, koncové podsítě potom až nejdelší možný – 64bitů. Zápis prefixu má následující formát:

IPv6_adresa/délka_prefixu

Délka prefixu řekne, kolik bitů od počátku adresy je považováno ze prefix. Příkladem budiž zápis:

1111:cafe:a123::1/48

Typy adres

Definovalo se několik typů adres, které sdružují adresy se společnou charakteristikou. Naprostá většina IPv6 adres spadá do skupiny globálních individuálních adres. Rozdělení adres lze vidět v následující tabulce:

Prefix	Význam
::/128	nedefinovaná adresa
::1/128	smyčka (loopback)
fc00:://7	unikátní individuální lokální adresy
fe80::/10	individuální lokální linkové adresy
ff00::/8	skupinové adresy
ostatní	individuální globální adresy

Tabulka 3: Rozdělení typů IPv6 adres

Globální individuální adresy jsou v celém Internetu jedinečné. Přidělují se hierarchicky tzn. nejdříve se přidělí prefix poskytovateli připojení k Internetu (lokální registr), který jeho části s delšími prefixy přiděluje svým zákazníkům. Tento přístup je velmi vhodný z hlediska směrování v Internetu, ve směrovacích tabulkách tak může být jediný záznam pokrývající celou tuto síť. V RFC 3578, byl definován jednoduchý model, kdy je každá adresa členěna na tři části:

- globální směrovací prefix – je dán délkou 48 bitů. Tento prefix přiděluje lokální internetový registr.
- identifikátor podsítě – má délku 16 bitů a slouží k rozlišení jednotlivých podsítí. Těch může být až 65536, tento rozsah už spravuje administrátor koncové sítě.
- identifikátor rozhraní – ten zabírá celou zbylou 64 bitovou polovinu celé adresy. Je to sice plýtvání, ale dle RFC 4291 je toto vyžadováno, spolu s použitím identifikátoru ve tvaru modifikovaného EUI-64.

Adresní prostor OU

Síť národního výzkumu Cesnet2 [9] má přidělen prefix 2001:718::/32, ze kterého vyčlenila pro počítačovou síť Ostravské univerzity tento prefix:

2001:718:1005::/48

V naší síti tak můžeme vyčlenit až 65536 podsítí s délkou prefixu 64. Na každé takové podsíti může být připojeno až (teoretických) $18*10^{18}$ rozhraní. Vzhledem k velikosti podsítí s nejdelším možným prefixem nemá význam přidělovat jednotlivým koncovým sítím větší rozsahy adres, než s prefixem 64. Proto bylo rozhodnuto přidělovat prefixy dle jednoduchého klíče

2001:718:1005:VLANID::/64

Adresa brány pak končí vždy 2001:718:1005:VLANID::1/64.

2.3 ICMPv6

Je to protokol zajišťující v sítích důležité funkce jako je ohlašování chybových stavů, testování dosažitelnosti obecně výměně některých provozních informací. Základ pro IPv6 je definován v RFC 4443: *Internet Control Message Protokol (ICMPv6) for the Internet Protocol version 6 (IPv6) Specification*, ale jeho rozšiřující části jsou definovány v dalších RFC dokumentech. O tomto protokolu považuji za nutné se zmínit pro jeho všeobecné využití prakticky při všech komunikačních činnostech v síti, ať už jde o směrování, přidělování adres či management sítě.

Následující obrázek ukazuje formát ICMP zprávy:

Typ	**Kód**	**Kontrolní součet**
Tělo zprávy		

Ilustrace 2: Hlavička ICMPv6 zprávy

První položka *Typ (Type)* rozlišuje, zda se jedná o chybovou zprávu (hodnoty 0 – 127) nebo o zprávu informační (128 – 255). Podtypy těchto zpráv jsou určeny *Kódem (Code)*. Obsah *Těla zprávy (Message body)* pak závisí typu zprávy. Přehled typů ICMP zpráv lze najít na adrese http://www.iana.org/assignments/icmpv6-parameters.

2.3.1 Chybové zprávy

Chybové zprávy se dělí na čtyři typy. *Typ* s **hodnotou 1** oznamuje nedosažitelnost cíle. Tento typ generuje směrovač v případě, že zpracovává datagram s cílovou adresou, na kterou není schopen datagram doručit. Důvod, proč tak nemůže učinit, zohlední v položce *Kód* (viz tabulka).

Kód	Význam
0	neznám žádnou cestu k cíli
1	správce zakázal komunikaci
2	mimo dosah zdrojové adresy
3	nedosažitelná adresa (cíl neodpovídá)
4	nedosažitelný port (cíl neodpovídá)
5	zdrojová adresa odporuje vstupně/výstupní politice
6	cesta k cíli je zakázána

Tabulka 4: Chybové zprávy ICMPv6

Pro ohlášení příliš velkého datagramu slouží ICMP zpráva s **hodnotou 2** v poli *Typ*. Dojde k tomu v případě, kdy má směrovač odeslat datagram linkou s nižším MTU než je velikost datagramu. V IPv6 může fragmentovat datagramy pouze odesílatel, nikoliv směrovač. Ten větší datagram zahodí, do odeslané zprávy pak přidá do pole *Tělo zprávy* hodnotu MTU, která problém způsobila. Odesílatel pak je schopen datagram fragmentovat na schůdnou velikost.

Při skončení doby platnosti datagramu (*Maximum skoků*) vygeneruje směrovač ICMP zprávu s **hodnotou** *Typu* **3**, kterou odešle odesílateli a datagram zahodí.

Zpráva *Typu* **4** pak značí, že příjemce obdržel chybný datagram. Tzn. že datagram obsahoval údaje, kterým příjemce nerozuměl. Ve zprávě je pak uveden počet bajtů od začátku datagramu, kde začíná položka s chybou.

2.3.2 Informační zprávy

Původně byly definovány pouze dvě. Výzva (echo) a Odpověď na výzvu (echo-reply). Využívají se programem *ping6* k testování dostupnosti rozhraní nějakého zařízení v síti. Zprávy obsahují 16ti bitové položky *Identifikátor* a *Pořadové číslo*. Ping6 pak odešle k testovanému cíli sekvenci zpráv se stejným identifikátorem, ale narůstajícím pořadovým číslem. Každý uzel v síti je pak povinen (dle norem) na žádost odpovědět.

V dalších dokumentech RFC pak byly definovány další typy informačních zpráv zabývající se např. skupinovým adresováním, ohlášením směrovače, podporou mobility nebo objevování sousedů. Také byly zavedeny experimentální zprávy zabývající se získáváním informací o uzlech.

2.4 Objevování sousedů

V originále *Neighbor Discovery*, je definováno v RFC 4861: *Neighbor Discovery for IP version 6*. Je to mechanismus sloužící zejména ke zjišťování linkových adres uzlů, hledání směrovačů, ověřování dosažitelnosti sousedů, zjišťování prefixů a parametrů sítě pro automatickou

konfiguraci i detekci duplicitních adres. K tomuto účelu využívá pět typů ICMPv6 zpráv a dvě zabezpečující tzv. SEND:

- výzva směrovači (router solicitation)
- ohlášení směrovače (router advertisemnt)
- výzva sousedovi (neighbor solicitation)
- ohlášení souseda (neighbor advertisement)
- přesměrování (redirect)
- žádost o certifikační cestu (certification path solicitation)
- ohlášení certifikační cesty (certification path advertisement)

3. Směrování

Obecně lze popsat směrování jako hledání cesty, kudy doručit datagram(y) k danému cíli. Směrování v sítích zajišťují směrovače, ovšem i každá stanice v síti musí vědět kam své pakety odeslat. Toto stanice (stejně tak směrovač) realizuje na základě informací uložených ve své *směrovací tabulce*.

3.1 Směrovací tabulka

Tato tabulka obsahuje informace o IPv6 prefixech (včetně délky prefixu) a informace o tom, přes který směrovač je cílová adresa dosažitelná. Samozřejmě pokud je cíl na stejné lince, odešle se datagram přímo. Pokud cílové adrese nevyhovuje žádný z definovaných prefixů, využije se v tabulce definovaná (pokud ji tedy někdo definoval, viz. níže) *implicitní cesta* (default route). Ta je označena v tabulce prefixem s nulovou délkou, konvenční zápis je **::/0**. Ve směrovací tabulce můžou být i záznamy pro jednotlivé adresy s délkou 128 (takovéto záznamy tvoří tzv. cache cílů, viz. kapitola 4.1.3).

Směrovací tabulka vzniká u koncových stanic na základě bezstavové automatické konfigurace, obdržením informací z DHCPv6 nebo se mohou záznamy staticky nakonfigurovat. U směrovačů je situace složitější, ty kromě statického přidávání záznamů, používají směrovací protokoly, které tyto záznamy doplňují, mění či mažou.

3.2 Statické směrování

Je směrování na základě staticky vytvořené směrovací tabulky. Záznamy v tabulce mění jen administrátor, v případě dynamicky přidělovaných informací síťovému rozhraní i DHCP server. Statické směrovací tabulky najdeme obvykle u koncových stanic se standardními záznamy pro loopback, vlastní prefix a implicitní cestu. U směrovačů se varianta statických tabulek využije především v případě známé, ne moc rozsáhlé sítě, kde jednotlivé podsítě mají jen jeden vstupní/výstupní směrovač.

Přidání implicitního záznamu do tabulky v OS Linux:

route -A inet6 add default gw 2001:718:1005:606::1

Přidání implicitního záznamu do tabulky v OS Windows 7:

netsh interface ipv6 add route ::/0 2001:718:1005:606::1

3.3 Dynamické směrování

Směrování, při kterém využívají směrovače pro tvorbu/úpravu svých tabulek *směrovací protokoly*. A to proto, že při rozsáhlejších, propojenějších a častěji se měnících sítích by ruční údržba směrovacích tabulek byla příliš náročná a pomalá. Směrovacími protokoly si tedy směrovače vyměňují informace o topologii síti a na jejich základě si pak své tabulky upravují.

Směrování se v Internetu organizačně člení na *autonomní systémy*, které jsou tvořeny skupinou sítí s jednotnou správou a směrovací politikou. Rozlišují se dvě skupiny směrovacích protokolů:

- IGP – směrovací protokoly využívané uvnitř AS, snažící se o rychlou reakci na změny. Zástupci jsou RIPng, OSPFv3 a IS-IS
- EGP – slouží k výměně směrovacích informací mezi AS. Tyto informace jsou velmi obsáhlé, reakce na změny je pomalejší. Jediným zástupcem je v této chvíli protokol BGP4+.

3.3.1 RIPng

Tento protokol je definován v RFC 2080: RIPng for IPv6. Protokol reaguje pomaleji na změny v síti a zná jen malou maximální délku cesty, avšak je jednoduchý a proto implementován v mnoha systémech. Z popsaného vyplývá, že je vhodnější pro méně rozsáhlé sítě, kde však z nějakého důvodu chceme dynamicky směrovat.

RIPng je založený na vektoru vzdáleností. Sítě (linky) propojující jednotlivé směrovače mají přiřazenu určitou cenu (v rozsahu 1 – 16, přičemž 16 znamená nedosažitelný cíl). Tu může definovat administrátor, nebo se častěji určí automaticky na základě přenosové rychlosti linky, zpoždění, zatížení atd. Když má směrovač rozhodnout, kudy vyšle paket, vybere cestu k cíli s nejmenším součtem cen linek, které k němu vedou. Směrovač, používající RIPng, má ve své směrovací tabulce uvedené u jednotlivých cílů i údaj o metrice (jak dlouhá je cesta k cíli). Tyto údaje si sousední směrovače mezi sebou vyměňují ve 30 sekundových intervalech, nebo když dojde ke změně směrovací tabulky a nebo jako odpověď na požadavek souseda. Požadavek na odpověď odešle přímo žadateli, v prvních dvou případech posílá informace ze své tabulky na skupinovou adresu ff02::9. Po obdržení těchto údajů si k nim směrovač přičte cenu linky, kterou informace přišla a údaje porovná se svou tabulku. Může pak cílové sítě přidat, ubrat a event. u cílů změnit metriky a adresu dalšího směrovače. RIPng potřebuje mí ve směrovací tabulce tyto údaje:

- prefix cíle (plus jeho délka)
- metriku celkové cesty k cíli
- adresu dalšího směrovač na cestě
- příznaky změny
- časové údaje (doba platnosti)

U přímo připojených sítí k směrovači samozřejmě chybí údaj o dalším směrovači na cestě, metrika je pak rovna přímo ceně linky.

Na následujícím obrázku je formát zprávy RIPng:

<table>
<tr><td colspan="2">Příkaz</td><td>Verze=1</td><td colspan="2">Rezerva=0</td></tr>
<tr><td rowspan="2">Položka 1</td><td colspan="4">Prefix</td></tr>
<tr><td colspan="2">Značka cesty</td><td>Délka prefixu</td><td>Metrika</td></tr>
<tr><td colspan="5">...</td></tr>
<tr><td rowspan="2">Položka N</td><td colspan="4">Prefix</td></tr>
<tr><td colspan="2">Značka cesty</td><td>Délka prefixu</td><td>Metrika</td></tr>
</table>

Ilustrace 3: Formát zprávy protokolu RIPng

V položce *Příkaz (Command)* lze nastavit hodnotu 1 pro požadavek na souseda, nebo hodnotu 2 signalizující odpověď či aktualizaci. Následují položky jednotlivých cílů. U každého je obsažen *Prefix*, *Délka prefixu (Prefix len)* a *Metrika (Metric)*. *Značku cesty (Route tag)* RIPng nevyužije, pouze předává informaci v této položce obsažené dále.

3.3.2 OSPFv3

Proti RIPng dokáže zajistit směrování i v rozlehlejších sítích a rychle reaguje na změny. Je definován v RFC 5340: *OSPF for IPv6*, a označován je jako verze 3.

Protokol je založen na stavu linek, kdy každý směrovač (s OSPF) v síti si udržuje aktuální mapu sítě. Pokud dojde ke změně, tak OSPF zařídí okamžité informování o této změně ostatní směrovače. Každý směrovač si pak z této mapy spočítá strom nejkratších vzdáleností ke všem

cílovým sítím, které zná. Kořenem tohoto stromu je směrovač sám. Zjistí tak kudy vedou nejkratší cesty k jednotlivým cílům a údaje si zapíše do směrovací tabulky.

Mapa sítě

Je pro OSPF orientovaný graf, kde vrcholy jsou směrovače a skupinové sítě (více než dva směrovače pro síť). Sítě mohou být označeny jako *koncové*, či jako *tranzitní* (to v případě, že přes ně prochází i data určená jiným cílům). Hrany grafu jsou ohodnoceny čísly v rozmezí 0 – 65535. Mluvíme tady o ceně cesty. Ta např. zohledňuje i přenosové rychlosti linek. Při hledání optimální cesty k cíli pak OSPF sčítá ceny linek a hledá variantu s nejmenším součtem.

Při výměně informací o stavu/změnách v mapě sítě si nejdříve směrovače automaticky zjišťují, které další směrovače mají ve svém okolí. To se provádí opakovaným vysíláním *Hello* paketů na všechna rozhraní, kterým se oznamuje přítomnost směrovače v síti. *Hello* paket obsahuje identifikátory všech směrovačů, o kterých odesílatel ví a také identifikaci *pověřeného směrovače* (designated router, viz níže). Směrovač si pak zjištěné směrovače zařadí do dvou kategorií:

- *Okolní směrovače* – všechny směrovače s přímým spojením, ať už dvoubodovou linkou nebo skupinovou.
- *Sousedé* – ti se vybírají z okolních směrovačů. V případě point-to-point linky se směrovače vždy stanou sousedy. V případě skupinové linky si nejdříve vyberou ze svého středu *pověřený směrovač* a s ním se pak stanou sousedy. Mezi sebou už pak ne (s výjimkou záložního pověřeného směrovače).

Verze=3	**Typ zprávy**	**Délka paketu**	
Identifikátor směrovače			
Identifikátor oblasti			
Kontrolní součet		**Identifikátor instance**	**0**

Ilustrace 4: Formát zprávy OSPFv3 protokolu

Mapa sítě tedy vzniká na základě oznámení, kterými si směrovače vyměňují informace o stavu sítě v určitém místě. Těmto oznámením se říká *LSA*. Existuje několik typů LSA a odesílají jen směrovače, u kterých informace o změně vznikla.

Synchronizace map dvou sousedů probíhá v několika krocích. Nejdříve pošlou sadu zpráv *Popis databáze (Database description)*, ve kterých jsou obsaženy identifikátory a verze LSA, tvořících jejich mapu sítě (alias databázi linek). Údaje porovnají a pokud naleznou doposud neznámé, nebo zastaralé LSA, požádají o ně pomocí *Žádosti o stav linky (Link state request)*. Odpovědí je jim pak *Aktualizace stavu linky (Link state update)* a po úpravě údajů v mapě jsou pak jejich pohledy na okolní síť stejné. Pokud dojde k nějaké změně v síti, směrovač ihned informuje

své sousedy zprávou Aktualizace stavu linky obsahující příslušné LSA. Sousedi pak toto LSA po úpravě své mapy šíří dále. Tomuto se říká záplavový algoritmus (*flooding*). Doručení aktualizace se potvrzuje zprávou *Potvrzení stavu linky (Link state acknowledgment)*.

Typ	Název	Význam
1	Hello	zjištění okolních směrovačů
2	Popis databáze	shrnutí obsahu databáze linek
3	Žádost o stav linky	vyžádání LSA
4	Aktualizace stavu linky	aktualizace (odeslání LSA)
5	Potvrzení stavu linky	potvrzení aktualizace

Tabulka 5: Typy OSPFv3 zpráv

Oblasti

Oblasti (area) byly pro potřeby směrování protokolem OSPF navrženy proto, aby tento protokol zvládal i velké sítě, kde by ale synchronizace map trvala příliš dlouho. Autonomní systém se tak dělí na jednotlivé oblasti a omezí se takto objem přenášených směrovacích informací mezi všemi směrovači v AS. Každá taková to oblast může být tvořena několika sítěmi a provozuje svou vlastní instanci směrovacího algoritmu a udržuje jen své mapy. Aktualizace map tak probíhá na mnohem menším prostoru a mezi méně směrovači, než by tomu tak bylo v celém AS.

Směrovač s rozhraními ve více oblastech se nazývá *hraniční směrovač (border router)*. Udržuje proto samostatné mapy sítě pro každou z oblastí a také pro každou nezávislou kopii směrovacího algoritmu. Všechny oblasti v AS jsou propojeny přes *páteřní oblast* s identifikátorem 0.0.0.0, vlastně všechny hraniční směrovače musí patřit i do páteřní oblasti. Datagram tak urazí cestu k cíli maximálně přes tři oblasti – jeho vlastní, páteřní a nakonec cílovou.

Směrovače tak mají detailní přehled o své oblasti, o ostatních pak mají jen rámcový přehled – ví jen jakou zvolit nejlepší cestu (hraniční směrovač ve své oblasti) podle cílové adresy. Tyto souhrnné informace o svých oblastech síří do jiných sítích hraniční směrovač. Činí to prostřednictvím LSA záznamů. V optimální případě se informace vtěsnají jen do jednoho LSA záznamu, např. o naší síti by náš hraniční směrovač posílal zprávu, že za ním leží síť s prefixem 2001:718:1005::/48. Tyto zprávy se dál síří v jednotlivých oblastech a místní zařízení se dozví, kudy posílat datagramy určené cílům v jiných oblastech.

Podobným principem se řeší i předávání cest k cílům v jiných autonomních oblastech. Hraniční směrovač AS, po obdržení takových to informací od externího směrovacího protokolu, přepošle toto sdělení v podobě externích LSA do páteřní oblasti. Z té pak putují přes hraniční směrovače do dalších oblastí s výjimkou oblastí, které jsou definované jako *koncové (stub area)*

(typicky oblasti s jedním hraničním směrovačem). Z takovéto oblasti se směrování za hranice AS provádí skrz implicitní cestu.

3.3.3 IS-IS

Definován je v RFC 1195: *Use of OSI IS-IS for routing in TCP/IP and dual environments*. Jde o směrovací protokol, který je ve verzi IPv4 využíván oproti OSPF velmi málo. V souvislosti s větším rozšířením IPv6 se začal nasazovat více, např. v evropské síti GÉANT2 [1]. Zejména to je z toho důvodu, že IS-IS bylo od počátku navrženo jako obecné pro libovolný síťový protokol a nezávisle na jeho adresách a jeho implementace ve světě IPv6 nebyla tak složitá.

IS-IS je protokol založený na stavu linek (stejně jako OSPF, který z IS-IS prakticky vychází), kdy si všechny směrovače udržují aktuální mapu sítě. Z ní pak vypočítají algoritmem hledání nejkratších cest svou směrovací tabulku. Případné změny v síti okamžitě směrovač začne šířit záplavovým algoritmem dále.

IS-IS také dělí AS na oblasti a udržuje kompletní mapy jen v jejich rámci. Oproti OSPF ale patří směrovač vždy celý do oblasti, hranice mezi oblastmi pak prochází linkami. Směrovače se dělí do dvou úrovní. Směrovače v úrovni 1 se starají o vnitřní topologii v oblasti, směrovače v úrovni 2 zajišťují komunikaci a výměnu informací mezi jednotlivými oblastmi. Spolu „se baví“ jen směrovače stejné úrovně. Meziúrovňovou výměnu informací pak zajišťují směrovače, které mají definované obě úrovně – značí se jako L1/L2 směrovače (analogie s L2/L3 přepínači?). V autonomním systému s IS-IS není definována žádná z oblastní jako páteřní. Páteř zde tvoří všechny L2 směrovače a prochází tak všemi oblastmi. Pokud je tedy potřeba směrovat datagram do jiné oblasti, L1 směrovač(e) jej předají L2 směrovači oblasti. L2 infrastrukturou se pak dostane do cílové oblasti, kde už jej L1 doručí adresátovi.

3.3.4 BGP4+

Rozšíření (nejen) pro IPv6 je definováno v RFC 4760: *Multiprotocol Extensions for BGP-4*. Je to externí směrovací protokol, skrz který se vyměňují informace mezi AS, na čemž v současné době stojí směrování v Internetu.

Směrovač s BGP shrne do jedné cesty všechny prefixy ze svého autonomního systému a ohlásí je svým sousedům v dalších AS. Ti je předávají dál a postupně tak ví každý směrovač, co je kudy dosažitelné. Ovšem směrovače si musí spočítat, která z možných cest k cíli je nejvhodnější. Sousedy si směrovač s BGP nehledá sám, ale konfiguruje je správce. S každým z těchto sousedů pak směrovač udržuje trvalé TCP spojení. Na úvod tohoto spojení si vymění veškeré směrovací informace, pak jejich aktualizace. Důležité je, že v případě přerušení tohoto spojení, směrovač prohlásí souseda za nedostupného a odstraní ze své směrovací tabulky cíle, které od něj získal.

BGP směrovací údaje uchovává ve třech bázích směrovacích informací (Routing Information Base, RIB):

- vstupní báze – v té jsou informace získané od sousedů. Ty se posuzují a na jejich základě se pak mění lokální báze.
- lokální báze – představuje směrovací tabulku
- výstupní báze – v té jsou informace, které směrovač ohlašuje sousedům

BGP používá několik typů zpráv, které mají shodný začátek. Obsahují *Znamení (Marker)*, obsahem jsou samé jedničky, následuje *Délka (Lenght)* zprávy a její *Typ (Type)*. Typy zpráv jsou v následující tabulce:

Typ	Určení
1=OPEN	zahájení vzájemné komunikace
2=UPDATE	zprávy o změnách ve směrování
3=NOTIFICATION	chybové hlášení
4=KEEPALIVE	udržování komunikace

Tabulka 6: Typy BGP zpráv

Nejdůležitější je zpráva UPDATE, která ohlašuje změny ve směrovacích tabulkách. Za výše uvedenými společnými položkami následuje pole *Zrušené cesty (Withdrawn routes)*, což jak název napovídá, obsahuje prefixy cest, které směrovač ohlašuje jako zrušené. Zbytek zprávy obsahuje informace o ohlašované cestě (ohlásit jde ve zprávě jen jedna cesta, zrušit jich jde vícero). Uvedeny jsou *Atributy cesty (Path attributes)*, které jsou popsány v následující tabulce:

Atribut	Význam
ORIGIN	říká odkud pochází informace (IGP, EGP, odjinud)
AS_PATH	seznam AS, kudy prošlo ohlášení o cestě
NEXT_HOP	adresa směrovače, který je první na cestě k cíli
MULTI_EXIT_DISC	slouží k rozhodování mezi několika cestami do téhož sousedního AS
LOCAL_PREF	posílá informaci o preferenci cesty, ale jen směrovačům ve svém AS
ATOMIC_AGGREGATE	informuje, že spojil několik cest do obecnějšího, kratšího prefixu
AGREGATOR	adresa a číslo AS směrovače, který cesty agregoval

Tabulka 7: Seznam Atributů cesty

Následuje seznam prefixů, které do cesty spadají (což je obvykle seznam všech prefixů z cílového AS), v položce *Informace o dosažitelnosti sítí (Network layer reachability information)*.

Když směrovači dorazí od souseda zpráva s aktualizací, upraví si informace v příslušné vstupní bázi. Následně tyto informace posoudí a vybere cesty pro svou lokální směrovací tabulku, cesty pro ohlášení sousedům ve stejném AS, cesty pro ohlášení sousedům v jiných AS a event. se rozhodne vhodné cesty agregovat a snížit tak objem směrovacích informací. Toto rozhodování probíhá ve třech fázích:

1) Spočítá se preference z každé vstupní báze (přiřadí cestě míru výhodnosti).
2) Dle vypočtených preferencí určí nejvýhodnější cestu pro každý známý cíl a zavede ji do lokální báze.
3) Naplní výstupní báze pro jednotlivé sousedy v závislosti na změnách v lokální bázi. Posílá jim jen cesty, které sám používá tzn. z jeho pohledu nejvýhodnější cesty k cíli.

3.4 Směrování v síti OU

V prvním cíli jsem si stanovil předělání směrování na centrální síťový prvek, kterým je v naší síti L3 přepínač Cisco Catalyst 6509. Původní hlavní směrovač pro IPv6 byl L3 přepínač Cisco Catalyst 3750. Důvody pro přesunutí byly sjednocení se směrováním na IPv4, větší výkon směrovače 6509 a také umístění firewall modulů (FWSM – firewall switch modul) v tomto stroji. Směrovalo se staticky a tento způsob jsem prozatím ponechal. Topologie počítačové sítě vychází z umístění jednotlivých budov Ostravské univerzity, které jsou především v centru Ostravy, ale i v odlehlejších částech města. Jen málo z těchto budov má více přístupových linek, proto je v tuto chvíli výhodnější statické směrování. Technické oddělení se ale snaží tlačit na vybudování více okruhů a v případě úspěchu už by nasazení jednoho ze směrovacích protokolů bylo na místě. Pak bych se zřejmě rozhodoval mezi variantami RIPng, z důvodu menšího počtu hopů v síti a jednoduchosti protokolu, a IS-IS kvůli jeho schopnosti pracovat současně s IPv4 i IPv6.

Postup nasazení

Jak z textu už vyplynulo, funkci centrálního směrovačem v síti OU plní Cisco Catalyst 6509. S ním jsou, většinou optickými linkami, spojeny L3 přepínače Cisco Catalyst 3750 na jednotlivých budovách fakult (podsítích). Na 3750 jsem zachoval nastavené IPv6 prefixy.

Změna ve statickém směrování proběhla v několika krocích:

- nastavení směrování a přidělení IPv6 adres rozhraním VLAN na 6509 (a zároveň odmazání adres z původního umístění na 3750)
- vytvoření spojovacích linek mezi centrální 6509 a lokálními směrovači 3750
- nakonfigurování směrovacích tabulek na 6509 a lokálních 3750

Nastavení 6509

Pro účely této práce jsem byl nucen upgradovat tento stroj na IOS (operační systém Cisca) verze 12.2(33)SXI3. Je to (v době psaní textu) nejnovější verze, mimo jiné s rozšířenější podporou IPv6. Více se tento bod ale dotýká DHCPv6, viz kapitola 4.3. Směrovaní IPv6 sítí samozřejmě plně podporovaly i starší verze IOS. Samotná konfigurace zařízení probíhá v tzv. konfiguračním módu, kam se přepne příkazem:

configure terminal

Dále je třeba globálně povolit IPv6 směrování a to příkazem:

ipv6 unicast-routing

Následovalo nastavení jednotlivých rozhraní VLAN. Ukázka nastavení na jedné z VLAN z pohledu IPv6 (nastavení ostatních VLAN se pak liší jen v identifikaci VLAN a přidělených prefixech):

interface vlan 607

ipv6 address 2001:718:1005:607::1/64

ipv6 enable

ipv6 nd prefix 2001:718:1005:607::/64 no-autoconfig

ipv6 nd managed-config-flag

ipv6 dhcp relay 2001:718:1005:606::10

Prvním příkazem z této skupiny se přepneme do konfiguračního módu rozhraní. V něm pak následuje přiřazení IPv6 adresy s délkou prefixu. Příkazem *ipv6 enable* se povolí na tomto rozhraní provoz IPv6 (nicméně v reálu stačí pouze přiřadit rozhraní IPv6 adresu). Příkazem *ipv6 nd prefix 2001:718:1005:607::/64 no-autoconfig* řekneme směrovači, aby do připojené linky ohlašoval jen tento prefix, ale se zákazem automatické konfigurace. Další dva příkazy mají souvislost s DHCPv6 a rozebírám je v kapitole 4.3.

Kromě nastavení rozhraní koncových sítí bylo důležité přiřazení IPv6 adresy na rozhraní spojovací linky do Cesnetu již známými příkazy:

configure terminal

interface vlanXXX

ipv6 address 2001:718:1000:10::2/64

ipv6 enable

Nastavení spojovacích linek mezi centrálním a lokálními směrovači

Postup nastavení rozhraní je podobný. Rozdílem je, že pro spojovací sítě využívám link-

local adres, které jsou dosažitelné jen v rámci linky samotné. Ukázka nastavení jedné ze spojovacích VLAN:

na 6509

configure terminal

interface vlan 250

ipv6 address fe80::1 link-local

ipv6 enable

na 3750

configure terminal

interface vlan 250

ipv6 address fe80::10 link-local

ipv6 enable

U dalších spojovacích sítí je jiný identifikátor VLAN, ale adresy jsou nastavené **vždy stejné** jako na ukázkovém příkladu. Toto nastavení adres umožňuje jejich dosah jen v rámci linky.

Nakonfigurování směrovacích tabulek

Na lokálních 3750 šlo jen o nastavení implicitní cesty, na 6509 pak o naplnění kompletní směrovací tabulky pro IPv6 síť OU.

na 3750

configure terminal

ipv6 route ::/0 interface vlan250 fe80::1

Po přepnutí do konfig. módu jsem nastavil implicitní cestu (::/0) ven ze sítě. Všechny datagramy určené pro jinou síť jsou jsou tedy odeslány rozhraním Vlan250 na adresu fe80::1.

na 6509

configure terminal

ipv6 route 2001:718:1005::/48 Null0

ipv6 route ::/0 2001:718:1000:10::1

ipv6 route 2001:718:1005:207::/64 Vlan250 fe80::10

ipv6 route 2001:718:1005:399::/64 Vlan251 fe80::10

...

První nastavenou cestou *ipv6 route 2001:718:1005::/48 Null0* zajistím zpracování paketů přiděleného adresního prostoru naší síti a vyvarujeme se tak vzniku smyček při zpracování datagramů směrujících do nepoužitých částí přiděleného adresního prostoru. Dále následuje implicitní cesta (::/0) vedená přes směrovač poskytovatele připojení a všechny cesty k sítím, které nejsou přímo připojeny k centrálnímu prvku. Další záznamy již specifikují konkrétní cesty - například záznam *ipv6 route 2001:718:1005:399::/64 Vlan251 fe80::10* říká, že všechny pakety určené cíli s prefixem 2001:718:1005:399::/64 se mají odeslat rozhraním Vlan251 na adresu fe80::10.

Funkčnost a správnost konfigurace jsem ověřil jednak výpisem směrovacích tabulek pro IPv6 - na obou typech směrovačů se tabulka vypíše příkazem s*how ipv6 route.* A dostupnost podsítí pak příkazem *ping ipv6 adresarozhranípodsítě*. Popis dalších možných nastavení a dostupných příkazů je v [4] a [5].

4. Přidělování IPv6 adres

Přidělování adres stanicím může ve světě IPv6 probíhat *stavově,* bezstavově a nebo, v případě serverů event. v jiných speciálních případech, je může být vhodnější nastavit IPv6 adresu staticky.

4.1 Statické přidělení adresy

V **linuxových** systémech se toto provede příkazem:

ifconfig eth0 add 2001:718:1005:207::100/64

Identifikátor rozhraní (eth0) a prefix se případ od případu liší. Druhou možností je nastavit parametry IPv6 v konfiguračním souboru, které po spuštění OS načtou startovací skripty. Na mé pracovní stanici se jedná o konfigurační soubor */etc/network/interfaces.* Parametry v něm uvedené mohou být následující:

iface eth0 inet6 static

address 2001:718:1005:606::169

netmask 64

Těmito nastavenými parametry se zajistí přidělení IPv6 adresy 2001:718:1005:606::169 s délkou prefixu 64 na rozhraní eth0. Z bezpečnostních i provozních důvodů, je vhodné vypnout automatickou konfiguraci adresy. Toho se dá docílit změnou jednotlivých parametrů v konfiguračních souborech umístěných v */proc/sys/net/ipv6/all* (nastavení v adresáři *all* platí pro všechna rozhraní; místo něj lze zvolit adresář např. *eth0*). Konkrétně nastavit hodnotu 0 v souborech *accept_ra* a *autoconf*. Více o této problematice v [8] a [15].

V případě **Windows 7/Vista/2008** nastavení adresy docílíme příkazem:

netsh interface ipv6 add address 5 2001:718:1005:606::130

Hodnota 5 vyjadřuje identifikátor rozhraní a může se lišit pro každý počítač.

Vypnutí automatické konfigurace adresy lze docílit příkazem:

netsh interface ipv6 set privacy state=disable

Informace o dalším možném nastavení jsou popsány v [13].

4.2 Bezstavové přidělování IP adres

Novým způsobem přidělování adres (proti IPv4) je tzv. bezstavové. Principem je u uzlu v síti automatické určení vlastní adresy. Tu si může určit na základě informací obdržených od směrovače, který v určitých časových intervalech vyšle tzv. *ohlášení směrovače (router advertisment)*. Rozšiřující informace ke kapitole 4.2 jsou dostupné v [1] a [2].

4.2.1 Ohlášení směrovače

Toto ohlášení **může**, prostřednictvím protokolu ICMP, vysílat každý směrovač v síti (samozřejmě s podporou a konfigurací IPv6) .

Typ=134	**Kód=0**				**Kontrolní součet**
Omezení skoků	**M**	**O**	**H**	**rezerva=0**	**Životnost implicitního směrovače**
Trvání dosažitelnosti					
Interval opakování					
Volby ...					

Ilustrace 5: Formát ICMP zprávy "Ohlášení směrovače"

Na obrázku je znázorněn paket ohlášení (ICMP). Ze základních parametrů bych zmínil *Životnost implicitního směrovače (router lifetime)*, který říká, jak dlouho bude směrovač fungovat jako implicitní pro danou síť (údaj v sekundách). Příznak *M* (*Managed address configuration*, stavová konfigurace adres) říká, že adresy a další komunikační parametry přidělí DHCPv6. Příznak *O* (*Other stateful configuration*, další parametry stavové konfigurace) rozhoduje o využití DHCPv6 i pro ostatní údaje sítě jako je např. adresa DNS serveru. Kombinace nastavených příznaků *M* a *O* jsou v následující tabulce.

M	O	význam
1	-	veškeré informace dodá DHCPv6
0	1	kombinace bezstavové konfigurace (adresa, prefix sítě, brána) a DHCPv6 (DNS, atd.)
0	0	nevyužívat DHCPv6

Tabulka 8: Kombinace příznaků M a O

Příznak *H (Home agent, domácí agent)* slouží jako podpora mobility a směrovač říká, že je schopen pracovat jako domácí agent. Časové údaje *Trvání dosažitelnosti (reachable time)* a *Interval opakování (retrans timer)* říkají, jak dlouho má být uzel považován za dosažitelný, respektive interval mezi výzvami. V části *Volby* směrovač sdělí po jedné volbě pro každý prefix IP adres, který se v dané síti používá (čili může být více prefixů v jedné fyzické síti), dále může ohlásit velikost MTU v síti či svou linkovou adresu. Na následujícím obrázku je volba *Informace o prefixu (prefix information)*.

Typ=3	**Délka=4**	**Délka prefixu**	**L**	**A**	**R**	**rezerva=0**
Doba platnosti						
Doba preferování						
rezerva=0						
Prefix						

Ilustrace 6: Zpráva "Informace o prefixu"

Parametr *Délka prefixu (prefix lenght)* určí, kolik bitů je platných z údaje v poli *Prefix*. Časové údaje (v sekundách) *Doba platnosti (valid lifetime)* a *Doba preferování (prefered lifetime)* udávají, jak dlouho prefix platí, respektive jak dlouho mají být preferované adresy vzniklé z tohoto prefixu (hodnota 0xffffffff znamená nekončící trvanlivost adresy). Příznak *L (on-Link)* – říká, že prefix lze využít k rozhodnutí, který uzel je lokální. Příznak *A* (*Autonomous address-configuration*, autonomní konfigurace adres) - říká, že prefix lze použít k automatické konfiguraci vlastní adresy (nenastavený příznak *A* zakáže bezstavovou konfiguraci). Příznak *R (Routed address)* – je-li nastavený, obsahuje položka *Prefix* kompletní globální adresu směrovače.

4.2.2 Určení vlastní adresy

Při automatickém určení vlastní adresy uzel začne tím, že si vytvoří lokální linkovou adresu, kdy ke standardnímu prefixu lokálních linkových adres fe80::/10 připojí identifikátor svého rozhraní. Uzel dále provede detekci eventuální duplicitní adresy tak, že použije *objevování sousedů*. Rozešle výzvu sousedům, ve které hledá možného vlastníka adresy, kterou si vygeneroval. Pokud dorazí ohlášení souseda, znamená to, že tento má stejný identifikátor rozhraní a automatická konfigurace uzlu nemůže pokračovat. Tento scénář je ale málo pravděpodobný, takže odezva na objevování by měla být negativní a uzel si vytvořenou lokální linkovou adresu přidělí.

Pokračovat v automatické konfiguraci může po získání informací o síti, ve které se nachází. Uzel proto musí počkat na *ohlášení směrovače*, event. o něj požádá tzv. *výzvou směrovači*. Z příznaků ohlášení se pak dozví parametry sítě. U každého z možných prefixů je uvedený příznak, zda pro něj má uzel použít bezstavovou konfiguraci adres. Pokud je tedy příznak *A* nastavený, tak uzel připojí k prefixu svůj identifikátor rozhraní a vzniklou IPv6 adresu si přidělí.

4.2.3 Konfigurace směrování

Uzly v síti se dokáží naučit i směrování, kdy si potřebné informace pro odesílání dat mohou udržovat v:

- *Cache cílů (Destination cache)* – obsahuje směrovací informace pro konkrétní cíle. Ke každému z těchto cílů je přiřazena první adresa po cestě k němu (klasický „next hop“).
- *Seznam prefixů (Prefix list)* – dle něj stanice posoudí, kdo je a není na stejné síti
- *Seznam implicitních směrovačů (Default router list)* – zde jsou informace o všech směrovačích, které se ve svém ohlášení deklarovali jako implicitní.

Tyto tři skupiny dat o směrování jsou deklarovány jako idea a v praxi se všechny dají reprezentovat směrovací tabulkou.

Postup odesílání datagramu je jednoduchý. Nejdříve se uzel podívá do cache cílů, zda se zde nenachází adresa příjemce. Pokud je, odešle datagram přímo na uvedenou adresu, pokud není, prohledá se seznam prefixů. V tomto se určí, zda se jedná o adresu lokální či vzdálenou. V případě lokální adresy se odešle paket přímo, v opačném případě se použije pro odeslání adresa příslušného implicitního směrovače.

Pokud uzel z nějakého důvodu zvolí pro odeslání datagramu nevhodný směrovač (nebo je cíl ve skutečnosti lokální), ten mu pošle nazpět ICMP zprávu o *přesměrování*. Tato zpráva obsahuje Cílovou adresu (Destination address) a Posílat přes (Target address), což znamená, že řekne odesílateli „ulož si do cache cílů adresu směrovače (nebo přímo cíle) pro tuto cílovou adresu“.

4.2.4 Konfigurace DNS serverů

Jak je patrné z výše popsaného, uzel si dokáže automatickou bezstavovou konfigurací nastavit IPv6 adresu rozhraní i jednoduchou směrovací tabulku. Co však sám neumí, je nastavit si adresu(y) DNS serveru. Dokument RFC 4339 říká, že jsou tři možnosti, jak toto nastavit. Já zmíním dvě z nich.

První z možností je nechat proběhnout bezstavovou konfiguraci a informace o DNS serverech doplnit stavově, k čemuž slouží nastavení příznaku *O* v ohlášení směrovače (viz kapitola 4.1.1). Při tomto nastavení sítě se využije tzv. bezstavové DHCPv6 definované v RFC 3736. Toto je jednoduchá verze DHCPv6 serveru, která používá jen 4 typy DHCP zpráv – žádost o informace, odpověď, předání žádosti a zprostředkování odpovědi (v případě DHCP relay). Klient odešle zprávu *Žádost o informace (Information request)*, jejíž součástí jsou informace, které klient od DHCP serveru požaduje. Server pak odešle *Odpověď (Reply)* s požadovanými daty.

Druhou, mnou zmíněnou, možností je doplnění informace o DNS serverech přímo do ohlášení směrovače. Tato možnost je popsána v RFC 5006. Toto rozšíření zavádí novou volbu v ohlášení a to *Rekurzivní DNS server (Recursive DNS server)*, která obsahuje položky *Délka* (z té se odvodí, kolik následuje adres DNS serverů), *Životnost* (v sekundách uvádí platnost adres) a pak samotné *Adresy DNS serverů*.

4.3 Stavové přidělování IP adres

Stavové přidělování lze označit jako **DHCPv6** (Dynamic host configuration protocol for IPv6), které je deklarováno v RFC 3315. Jedná se o známou situaci z původní verze 4 probíhající v několika krocích, kde host vyslal broadcast s žádostí o přidělení IP adresy. DHCP server (event. přes zprostředkovatele) pak klientovi odpověděl přímo nebo přes relay, která mu žádost o adresu přeposlala. V odpovědi jsou údaje jako IP adresa, prefix sítě, brána sítě, dns server(y), wins server(y), doménové jméno event. odkaz na bootovací obraz uložený na tftp serveru atd. Stejně tak účastníky přidělení IP adresy jsou klienti, zprostředkovatelé (relay) a DHCP servery. Relay a servery se souhrnně označují jako *agenti*. Nicméně nová verze má zákonitě odlišnosti, které je třeba v následujících kapitolách vysvětlit (doplňující informace jsou k dispozici v [1] a [2]).

4.3.1 Identifikátory

Původním identifikátorem klienta ve verzi 4 je ethernetová (tzv. MAC) adresa. Ve verzi 6 se zavádí identifikátor **DUID**, který by měl být stálý a nemusel by se změnit ani při výměně síťové karty . DUID má každý klient i agent. Autoři protokolu definovali způsoby, jak jej lze vytvořit. Prvním ze způsobů je vytvoření DUID z výrobního čísla zařízení (při předpokladu jeho přidělení výrobcem) a doménou výrobce. Takto vzniklý DUID by se neměl změnit po celou životnost klienta. Další variantou je kombinace linkové adresy a času jejího vytvoření. Takto vzniklý DUID ale vyžaduje trvanlivou zapisovatelnou paměť, kam se může jeho hodnota uložit. Poslední

možností pro určení DUID je využití samotné linkové adresy, bohužel tento DUID se při výměně síťové karty změní.

Druhou konstrukcí je tzv. **IA** *(identifikační asociace)*, což je několik konfiguračních informací přidělených jednomu rozhraní + jednoznačný identifikátor (IAID). Tento pak přidělí klient každému rozhraní, na kterém chce využít DHCP.

Shrnuto, DUID je identifikátorem klienta a IA identifikátorem jeho rozhraní.

4.3.2 Proces získání adresy

Jak jsem se už zmínil, tento proces je podobný tomu ve verzi 4. Klient vyšle žádost o přidělení potřebných informací o síti, dostane nabídku(y), vybere si a pak od DHCP serveru informace přijme. V předešlé verzi klient vyslal požadavek všesměrově (tzv. broadcast), ve verzi 6 jsou pak definované standardní skupinové adresy:

- ff02::1:2 – pro všechny DHCP agenty a servery
- ff05::1:3 – pro všechny DHCP servery

Typy zpráv používané DHCPv6 uvádím v následující tabulce:

1	výzva (solicit)
2	ohlášení serveru (advertise)
3	žádost (request)
4	potvrzení (confirm)
5	obnovení (renew)
6	převázání (rebind)
7	odpověď (reply)
8	uvolnění (release)
9	odmítnutí (decline)
10	rekonfigurace (reconfigure)
11	žádost o informace (information request)
12	předání (relay forward)
13	zprostředkovaná odpověď (relay reply)

Tabulka 9: Zprávy DHCPv6

Klient tedy vyšle na adresu ff02::1:2 *výzvu*, ve které obsažen jeho DUID, všechny IA i lokální linková adresa, kterou si přidělil. Pokud je DHCP server na stejné lince, odpoví *ohlášením*

serveru (přibalí i parametry vhodné pro jednotlivé IA) přímo klientovi na jím uvedenou linkovou adresu. Pokud je DHCP server na jiné síti, musí využít klient služby *zprostředkovatele*. Ten by měl mít nakonfigurován seznam serverů, kterým má požadavky na přidělení adres předávat (může být definována i obecná skupinová adresa ff05::1:3). Pře-poslaný dotaz zabalí do nové zprávy typu *předání*, odpověď pak server zabalí do *zprostředkované odpovědi*. Klient si pak se všech příchozích ohlášení vytvoří seznam DHCP serverů (já osobně preferuji jen jeden aktivní DHCP server + záložní). V další fázi klient vyšle zprávu *žádost*, ve které je uveden DUID DHCP serveru, opět na obecnou adresu agentů. K serveru žádost dorazí buď přímo, nebo přes relay. Ten ji posoudí a podle sítě, ze které požadavek vzešel a DUID klienta vybere adresu(y), které klientovi oznámí ve zprávě *odpověď* (event. *zprostředkovaná odpověď*). Klient si adresu ověří detekcí duplicitních adres a v případě konfliktu vyšle DHCP zprávu *odmítnutí*.

4.3.3 Obnovení adresy

DHCP server deklaruje, mimo jiné, i časovou platnost adresy. Po vypršení této lhůty klient požádá zprávou *obnovení* o prodloužení platnosti své adresy. Pokud mu z nějakého důvodu původní DHCP server neodpovídá, vyšle žádost o prodloužení platnosti ve zprávě *převázaní* všem dostupným serverům (tedy v případě, že další servery má ve svém seznamu).

Při ukončení své činnosti (vypnutí) by měl uzel informovat server zprávou *uvolnění* o uvolnění své IPv6 adresy pro další použití. V situaci, kdy se např. dočasně odpojí od sítě nebo se probudí po úsporném režimu, vyšle všem agentům zprávu *potvrzení*, ve které jsou parametry jeho IA. Pokud jsou údaje správné, příslušný server odešle *odpověď* s potvrzením, v opačném případě s odmítnutím.

V případě nějakých změn na síti může odeslat DHCP server zprávu rekonfigurace každému z klientů, kterých se tato změna týká. Pak proběhne kolečko zpráv mezi klientem a serverem s požadavkem na *obnovení* a *odpovědí*.

4.4 Řešení přidělování adres v síti OU

Při svém řešení jsem se rozhodl pro změnu přidělování IPv6 adres z bezstavového na stavové. K tomuto rozhodnutí mě vedla potřeba získání většího přehledu o uzlech pracujících na IPv6 síti a automatické nastavení adresy DNS serveru na stanicích. Tato změna ale měla bohužel pár úskalí, která zmíním dále v textu.

DHCPv6 server

Prvním úkolem této změny bylo vybrat a zprovoznit samotný DHCPv6 server. Z praktického hlediska se jevila jako nejrozumnější varianta spravovat DHCP server pro IPv4 i IPv6 na jednom fyzickém serveru. Vybírali jsme tedy, spolu se správcem serveru, mezi variantami Dibbler [14] a ISC DHCP [10]. Pro Dibbler hovořila lépe zpracovaná dokumentace i fakt, že je i součástí některých distribucí Linuxu. Pro ISC DHCP hovořila tradice tohoto serveru, firmy stojící

za ISC (záruka dlouhodobé podpory) a zejména dlouhá zkušenost s provozováním ISC DHCP pro IPv4. Volba tedy padla na server od ISC (i když k Dibblerovi jsem se oklikou částečně vrátil, viz odstavec o DHCP klientech).

Stabilní verze od ISC, která podporuje i IPv6, je nyní 4.1.1. Tento balík ovšem není součástí distribuce, proto bylo potřeba stažený balík ručně přeložit. Pro provoz DHCP serveru současně pro oba protokoly, je třeba spustit dvě instance. Jednu pro IPv4:

/usr/local/dhcp4/sbin/dhcpd -4 -cf /etc/dhcpd.conf -lf /var/lib/dhcpd/dhcpd.leases

a druhou pro IPv6:

/usr/local/dhcp4/sbin/dhcpd -6 -cf /etc/dhcpd6.conf -lf /var/lib/dhcpd/dhcpd6.leases

Přepínače -4 a -6 netřeba vysvětlovat, přepínač -cf odkazuje na konfigurační soubor a přepínač -lf na soubor, kam se ukládají leases (zapůjčené adresy).

Dále bylo potřeba nakonfigurovat dhcpd6.conf; následuje výpis jeho části:

```
authoritative;
default-lease-time 28800;
max-lease-time 28800;
get-lease-hostnames on;
log-facility local6;
option dhcp6.name-servers 2001:718:1005:601::10, 2001:718:1005:601::11;
option dhcp6.domain-search "osu.cz";
#
#  dhcp demon vyzaduje znalost podsite, ve ktere pracuje
#
subnet6 2001:718:1005:601::/64 {
}
#
# vlan 607
#
subnet6 2001:718:1005:607::/64 {
        range6 2001:718:1005:607::1001 2001:718:1005:607::ffff;
}
```

Při konfiguraci hovoříme o globálních parametrech, platných pro všechny podsítě, a o lokálních parametrech platných v rámci podsítě, skupiny či jen pro jednoho hosta. Deklarování položky *authoritative* řekne serveru, že může posílat DHCPNAK zprávy žadatelů, kteří tak můžou korektně dokončit proces získání adresy. Tato položka slouží jako ochrana před nějakou náhodnou instalací DHCP serveru v síti. Pokud klient nežádá o specifický čas platnosti své zápůjčky, položka *default-lease-time* mu dodá tento čas v sekundách (v IPv6 hovoříme o *valid lifetime* pro lease). *Max-lease-time* uvádí maximální možný čas platnosti adresy. Zapnutím get-lease-hostnames by měl DHCP server zjišťovat (a ukládat) jména zapůjčených adres. Dále následuje nastavení logování a poté dvě položky option. Položka *dhcp6.name-servers* dává klientům informaci o DNS serverech v síti a *dhcp6.domain-search* poskytuje informaci o implicitní doméně.

Deklaraci jednotlivých podsítí demonstruji na jedné z nich (konkrétně VLAN607). Podsíť se uvede definováním *subnet6*, za kterým následuje prefix dané podsítě. Uvnitř oddílu subnet6 jsem deklaroval rozsah přidělovaných IPv6 adres příkazem *range6* a uvedením první a poslední přidělované adresy. Poněkud zvláštní úlohu hraje část *subnet6 2001:718:1005:601::/64 { }*. Jedná se o prefix sítě, na které poslouchá rozhraní DHCPv6 serveru. Bez této deklarace nechtěl server nastartovat.

DHCPv6 zprostředkovatel (relay)

Aby žádosti klientů z jiných podsítí (než je ta s DHCP serverem) dorazily až požadovanému cíli, bylo nutné zprovoznit DHCP zprostředkovatele. Tuto funkci jsem nastavil na rozhraních směrovačů pro jednotlivé linky. Cisco Catalyst 6509 funkci dhcp zprostředkovatele ale podporuje až od řady IOS 12.2.33-SXI. V předešlé řadě SHX podporovali jen funkci dhcp serveru a klienta. Samotný upgrade IOS není není nic složitého, komplikace ovšem způsobilo Cisco svou změnou politiky poskytování svých IOS. Za pomoci třetí strany se problém vyřešil a po upgrade IOS už zbývalo definovat cíl zprostředkovávání. Konfigurace rozhraní tedy vypadá takto:

interface vlan 607

ipv6 address 2001:718:1005:607::1/64

ipv6 nd prefix 2001:718:1005:607::/64 no-advertise

ipv6 nd managed-config-flag

ipv6 dhcp relay destination 2001:718:1005:601::10

První dvě položky definují rozhraní a jeho prefix. Další už mají co do činění s přidělováním adres. Příkaz *ipv6 nd prefix 2001:718:1005:607::/64 no-advertise* řekne směrovači, že má do linky ohlašovat tento prefix, ale že klienti **nemají** pro tento prefix použít automatickou konfiguraci adresy. Dalším příkazem směrovač do linky ohlásí stanicím, že mají využít k nastavení adresy DHCPv6. Ti pak začnou proces komunikace s DHCP serverem (kapitola 4.2.2). Příkaz *ipv6 dhcp relay destination 2001:718:1005:601::10* pak směrovači říká, že má žádosti o přidělení IPv6

adresy přesměrovat na adresu DHCP serveru 2001:718:1005:601::10. Další možnosti nastavení jsou popsány v [4] a [5].

Dlouhou dobu visel otazník nad sítěmi směrovanými Catalysty 3750, jejich IOS totiž nepodporoval žádné DHCPv6 funkce. V jedné z novějších verzí pak podporoval funkce DHCPv6 server a klient. Teprve nedávno pak byl proveden upgrade na verzi 12.2.53(SE2), která už podporuje i DHCPv6 relay. Konfigurace zprostředkovatele na podsíti je pak stejná jako u boxu 6509.

DHCPv6 klient

Je naštěstí součástí všech novějších operačních systémů. Windows 7 a Vista jej mají implicitně zapnutý (tzn. že po instalaci systému je funkční podpora IPv6). Takže po zapojení počítačů s těmito OS do podsítě nakonfigurované pro využití DHCPv6, dostanou tyto stroje přidělenu od serveru IPv6 adresu a můžou začít komunikovat. U stále velmi rozšířených Windows XP je situace horší. Za prvé nemají po instalaci systému nainstalovanou podporu pro IPv6. Toto ale není problém rychle napravit. Pokud se jedná o XP se servisním balíkem 2 (SP2), stačí v příkazové řádce napsat *ipv6 install*. I po zprovoznění podpory pro IPv6 ale nemají XP DHCPv6 klienta a je nutno si pomoci softwarem třetí strany. Nevím zda existuje více klientů, ale já narazil jen na známého Dibblera [6] [14]. Po stažení kompletního instalačního balíčku (aktuálně verze 0.7.2) a spuštění instalace, je vhodné (z hlediska možných komplikací v síti spíše nutné) ponechat volbu instalovat jen DHCP klienta (nikoliv relay a server) a dokumentaci. Dále je třeba upravit konfigurační soubor klienta, který se nalézá v *Všechny programy → Dibbler ->→ Client Edit config file*. Tento soubor je již přednastaven a pro fungování v síti OU stačí odkomentovat následující:

```
iface „ Local Area Connection“ {
        ia
        option dns-server
        option domain
}
```

Místo *Local Area Connection* je pak třeba dosadit identifikátor IPv6 rozhraní, který se zjistí příkazem *ipv6 if* (v mém případě se jednalo o Rozhraní 5, přičemž dosadit stačí *iface 5*). Po této úpravě již zbývá jen nainstalovat službu klienta kliknutím na *Všechny programy → Dibbler → Client Install as service*.

U linuxových stanic slouží k získání IPv6 adresy z DHCP serveru démon **dhcp6c**. Pro potřeby v síti OU jej stačí spustit s parametry:

```
dhcp6c -d -c dhcp6c.conf
```

Volba -d umožní zapsat debugovací zprávy při procesu získávání adresy, přepínač -c odkazuje na konfigurační soubor. V tomto konfiguračním souboru (ideální je použít už vytvořený dhcpv6.conf)

pak stačí definovat:

interface eth0 {ttprefer-life-time 18000; trequest domain-name-servers;};

Tato jednoduchá deklarace říká, že dhcpv6 má žádat o adresu na rozhraní eth0 s požadavkem na získání adresy DNS serveru a preferovanou dobu zapůjčení 18000 sekund.

5. DNS

DNS překládá IP adresy na jména (uspořádaná hierarchicky) a naopak. Toto usnadnění práce v počítačové síti (de facto práci s Internetem obecně) vynikne ve světě IPv6 ještě více než v původní IPv4. Důvodem je samozřejmě větší délka, ale i hexadecimální zápis, IPv6 adresy. Po určitém, ne zrovna krátkém, vývoji je DNS pro IPv6 definován v dokumentu RFC 3596. DNS server dokáže předat informace o IPv6 i přes IPv4 síť a naopak.

5.1 Dopředné dotazy

Jedná se o dotazy na zjištění adresy k danému jménu. V IPv4 je tento záznam označován jako A (address record), v IPv6 pak (celkem logicky, při čtyřnásobné délce adresy) jako AAAA. Například záznam pro mé pracovní PC vypadá v DNS serveru takto:

radar AAAA 2001:718:1005:606::169

V případě podsítí, které používají více prefixů je třeba navýšit ekvivalentně počet AAAA záznamů u jednotlivých uzlů. Záznam pro mé PC by pak vypadal např. takto:

radar AAAA 2001:718:1005:606::169

2001:718:1005:607::169

5.2 Zpětné dotazy

Jedná se o dotazy na zjištění jména k dané IP adrese. Záznam v DNS se označuje ve verzi IPv6, stejně jako ve verzi IPv4, PTR. Dotaz se pokládá prostřednictvím doménového jména, které vznikne obrácením hexadecimálních číslic adresy a připojením domény ip6.arpa. V záznamu se nesmí zapomenout na vynechané nuly, např. PTR záznam v reverzní zóně vypadá pro můj počítač takto:

9.6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.6.0.5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa PTR radar.osu.cz.

Bohužel reverzní domény jsou u velkých sítí díky dlouhým IPv6 adresám značně rozsáhlé.

5.3 Řešení DNS v síti OU

Na serveru s distribucí Linuxu (a podporou IPv6) je nainstalován BIND verze 9 [11]. Tento server funguje pro překlad jmen a IPv4 adres. Aby mohl server odpovídat i na dotazy pro překlad IPv6 adres je třeba mít nastaveno v části *options* konfiguračního souboru named.conf:

listen-on-v6 { 2001:718:1005:601::10; };

V named.conf jsou nastaveny další důležité parametry (vypíšu jen ty, které mají souvislost s IPv6):

allow-recursion

{

2001:718:1005::/48;

};

- povolí odpovídat na rekurzivní dotazy jen z našeho prefixu.

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa." IN {

type master;

file "named/named.ip6.local";

allow-update { none; };

};

zone "osu.cz" IN {

type master;

file "/var/named/osu.cz/osu.cz";

};

zone "5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa" {

type master;

file "/var/named/osu.cz/ipv6.rev";

};

- deklarování potřebných zón.

V síti Ostravské univerzity byly prozatím nastaveny záznamy v DNS jen pro zařízení se

statickými IPv6 adresami uzly (typicky servery a administrátorské stanice). Definování jmen i pro hosty s adresami přidělenými DHCP serverem je mým úkolem do budoucnosti. Předpokládám realizaci pomocí dynamického DNS.

Nastavení zónového souboru pro dopředný překlad (část):

```
;osu.cz
$TTL        86400
@           IN    SOA    oudec.osu.cz.   hlavka.osu.cz. (
                  2010050101    3600     1800 2592000 86400 )
                  NS     oudec.osu.cz.
                  NS     albert.osu.cz.
                  NS     ns.ces.net
osu.cz.           MX     1 mailer.osu.cz..
                  MX     10 stu2.osu.cz.
oudec             A              195.113.106.10
                  AAAA           2001:718:1005:601::10
rampa             AAAA           2001:718:1005:601::5
neo               AAAA           2001:718:1005:606::168
radar             A              195.113.106.169
                  AAAA           2001:718:1005:606::169
gw6-607-bone      AAAA           2001:718:1005:607::1
```

Nastavení zónového souboru pro zpětný překlad (část):

```
;5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa
$TTL        86400
@           IN    SOA    oudec.osu.cz.   hlavka.osu.cz. (
                     2010042002     3600  1800 2592000 86400 )
;
                  NS     oudec.osu.cz.
                  NS     albert.osu.cz.
                  NS     ns.ces.net.
```

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.7.0.6.0.5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa. *PTR* *gw6-607-bone.osu.cz.*

0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.6.0.5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa. *PTR* *oudec.osu.cz.*

5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.6.0.5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa. *PTR* *rampa.osu.cz.*

8.6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.6.0.5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa. *PTR* *neo.osu.cz.*

9.6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.6.0.5.0.0.1.8.1.7.0.1.0.0.2.ip6.arpa. *PTR* *radar.osu.cz.*

Zónové soubory začínají vždy záznamem typu SOA, který obsahuje jméno primárního DNS serveru, emailovou adresu správce (místo zavináče je tečka), sériové číslo a časové údaje refresh, retry, expire a TTL. Záznamy NS odkazují na autoritativní DNS servery a MX na servery pro příjem pošty (včetně priority). Poté jsem už definoval dopředné (AAAA) a zpětné (PTR) záznamy, pro jednotlivé počítače [7].

6. Zabezpečení IPv6 sítě

Jak říká klasik, zabezpečení počítačové sítě je nikdy nekončící proces. Jaký je rozdíl v zabezpečení proti světu IPv4? V principu velmi podobný, nadále budou firewally, IPS systémy, šifrování, VPN přístup atp. V této kapitole postupně rozeberu rozšíření zabezpečení implementované přímo v IPv6 - IPsec a dále nastavení přístupových pravidel do naší sítě. Součástí zabezpečení počítačových sítí jsou i jasně deklarovaná pravidla pro chování na síti, ale ty platí stejně jak v původní IPv4 tak i v IPv6.

6.1 IPsec

Poslední verze je definována v RFC 4301: *Security Architecture for the Internet Protocol*. Teorie i praktické nasazení IPsecu není dle mého triviální věcí, a pro plné vysvětlení by bylo zapotřebí samostatné práce. Proto se budu snažit popsat jen základní principy a prostředky.

IPsec poskytuje dvě služby – autentizaci a šifrování. Při autentizaci se ověřuje pravost odesílatele datagramů, při šifrování se pak přidá zakódování obsahu, který by měl dešifrovat jen určený příjemce. Pro tyto účely se využívají rozšiřující hlavičky *AH* a *ESP*, nicméně od využívání AH se upouští, protože ESP plní tu samou funkci s přídavkem šifrování. Zabezpečený provoz může fungovat ve dvou režimech. V *transportním* jsou tyto rozšiřující hlavičky přímo součástí datagramu, v *tunelujícím* se pak celý původní datagram zabalí do nového datagramu obsahujícího bezpečnostní hlavičku. Pilířem provozu IPsecu jsou *bezpečnostní asociace* (a jejich správa), které určují jaké se použijí algoritmy, s jakými parametry a klíči, dobu platnosti spojení atd. Je to vlastně virtuální spojení dvou počítačů, které zajišťuje zabezpečený přenos dat. Tyto asociace lze spravovat manuálně nebo lze využít protokolu *IKEv2*.

Centrální směrovač Cisco 6509 podporuje vytvoření IPv6 IPsec tunelu *site-to-site* (spojení mezi sítěmi šifrované, uvnitř sítí už datagramy nešifrované). Vyžít tento zabezpečený tunel můžeme v případě spojení s pracovišti mimo naší síť (např. Fakultní nemocnice Ostrava) nebo se spolupracujícími subjekty (např. Západočeská univerzita v Plzni).

6.2 Nastavení přístupových pravidel (firewall)

Zabezpečení provozu na IPv4 sítích firewallem (bezpečnostní bránou) je dnes samozřejmostí a na sítích IPv6 by tomu nemělo být jinak. Není důvod, aby se nastavení přístupových pravidel pro přístup do sítě (podsítí) IPv6 lišilo od těch pro IPv4. Výjimkou je ale protokol ICMPv6, který je využíván v IPv6 při některých zásadních funkčních záležitostech (mobilita, objevování sousedů). Vznikl dokument RFC 4890: *Recommendations for Filtering ICMPv6 Messages in Firewalls*, který popisuje základní typy útoků pomocí ICMPv6 a zároveň obsahuje doporučení, jak se k určitým typům zpráv postavit. Zprávy dělí do pěti kategorií:

- propustit! – tyto zprávy musí firewallem projít, jinak může být narušena funkčnost sítě
- propustit – pokud není speciální důvod těmto zprávám bránit, je lepší je propouštět
- rozhodnout – záleží čistě na administrátorovi, jak se zprávami naloží
- zahodit – zprávy by neměly procházet
- netřeba – zprávy nebudou předávány dál, čili není nutno tvořit pravidla

Problémy můžou firewallům způsobovat zřetězené hlavičky, kdy potrvá delší dobu, než bude datagram zpracován nebo šifrované datagramy (ESP), kdy rozšiřující hlavičky vůbec neuvidí! Částečným řešením je důsledné nasazování lokálních (personal) firewalů na serverech i stanicích. Na OS na Linuxu toto řeší **ip6tables**. Například nastavení přístupových pravidel na DNS serveru by mohlo vypadat takto:

ip6tables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT

ip6tables -A INPUT -i eth0 -p tcp -m tcp --dport 53 -j ACCEPT

ip6tables -A INPUT -i eth0 -p udp -m udp --dport 53 -j ACCEPT

ip6tables -A INPUT -i eth0 -p ipv6-icmp -j ACCEPT

ip6tables -A INPUT -i eth0 -j DROP

Těmito příkazy jsou postupně přidány do tabulky INPUT pro rozhraní eth0 pravidla pro povolení paketů, které jsou odpovědí na odeslané pakety serveru, přistupují na tcp a udp port 53 (domain) a všem ICMPv6 zprávám. Veškeré ostatní pakety se zahodí.

Stanice s Windows 7 a Vista mohou využít svůj integrovaný firewall, kde se dají pravidla

pro IPv6 nastavit buď přes grafické rozhraní nebo příkazy network shellu *netsh advfirewall* (existuje i sada příkazů *netsh firewall*, ale Microsoft je důrazně doporučuje používat advfirewall) [13]. Stanice se staršími verzemi OS (XP, 2000) se musí spolehnout na personální firewally jiných výrobců.

Centrální firewall

Jako přístupový filtr do sítě (i v rámci sítě) využíváme karty *FWSM* umístěné v Catalystu 6509. Současná verze OS podporuje IPv6 jak z hlediska adresace a směrování (statického), tak z hlediska filtrace [3]. Jednotlivé sady pravidel jsou u Cisca nazývány *access-listy*, které se pak přiřazují k jednotlivým rozhraním. Přidání jednoho pravidla se děje v konfiguračním módu a má následující syntaxi:

ipv6 access-list id [line num] {permit | deny} protocol source [src_port] destination [dst_port]

Id značí identifikátor access-listu (číslo), *line* parametr říká na kterou pozici v access-listu pravidlo umístit, poté se definuje zda, se jedná o pravidlo, které provoz povoluje (*permit*) nebo zakazuje (*deny*). V položce *protocol* se definuje typ posuzovaného protokolu (ipv6, icmpv6, tcp, udp atd.). V případě protokolů vyšší vrstvy se kromě zdrojové (*source*) a cílové (*destination*) IPv6 adresy můžou definovat i zdrojové (*src_port*) a cílové (*dst_port*) porty. Celý access-list se pak přiřadí na příslušné rozhraní příkazem:

access-group access_list_name {in | out} interface if_name

U pravidel tedy určíme i v jakém směru mají být nasazena, jestli v příchozím do zvoleného rozhraní, nebo v odchozím.

Vzhledem k poměrně velkému množství pravidel, se pro jejich správu v naší síti využívá management softwaru *ASDM*. Bohužel současná verze ASDM **ignoruje** veškeré nastavení IPv6. Bez toho management softwaru je náročnost administrace FWSM velká a proto v tuto chvíli bylo produkční nasazení firewallu i pro IPv6 odloženo a budeme trpělivě očekávat verzi s podporou IPv6. Tento fakt má bohužel nemalý vliv na zatím opatrnější nasazování IPv6 na produkčních serverech (viz Závěr).

Přesto je vhodné alespoň základní ošetření přístupu do IPv6 sítě OU provést, neboť se skupina uživatelských stanic s IPv6 rozrůstá. Toto omezení přístupu lze nastavit pomocí známých access-listů na směrovači 6509 [4] [5]. Vzhledem k malému provozu po IPv6 síti a malému počtu pravidel tuto funkci zatím zvládne. Nastavení access-listu **by mohlo vypadat** takto:

ipv6 access-list INPUTipv6

permit tcp any any established

permit icmp any any

permit tcp any host 2001:718:1005:601::10 eq domain

permit udp any host 2001:718:1005:601::10 eq domain

permit ipv6 any host 2001:718:1005:606::168

deny ipv6 any any log

V globálním konfiguračním módu se definuje access-list s názvem INPUTipv6. V konfiguračním módu access-listu se pak přidají jednotlivá pravidla. Prvním příkazem se povolí přístup všem už navázaným spojením, dalším se povolí přístup ICMPv6 paketům a následující dvě pravidla povolí dotazy na DNS server po tcp i udp. Další pravidlo povolí přístup na testovací server a posledním se zakáže veškerý další přístup po IPv6, zahozené pakety se zapíší na logovací server. Zbývá nasadit vytvořený access-list na vstupní rozhraní do sítě:

interface vlan XXX

ipv6 traffic-filter INPUTipv6 in

7. Mobilita

Tento termín značí proces, kdy má mobilní zařízení (využívající IPv6) definovánu nějakou domovskou síť a při cestování po světě (přesněji po jiných sítích) využívá ke komunikaci svou registrovanou domácí IPv6 adresu. Domácí IPv6 adresa je pevná a měla by mít DNS záznam. Přes tuto adresu je pak mobilní zařízení dostupné, byť se vyskytuje na jiné síti se zcela jiným prefixem. Aby zařízení mohlo být dostupné, hlásí se k tzv. *domácímu agentovi*, obvykle směrovači dané sítě. Obecně pak může takovýto agent předávat tunelem datagramy směrované k domácí adrese zařízení na jeho současnou adresu. Mobilní zařízení však může využít optimalizaci cesty, kdy se snaží sdělit odesílateli svou současnou adresu a urychlit tak komunikaci. Současně uleví od provozu domácímu agentovi. Kromě jednotlivých zařízení, lze definovat i celou mobilní síť. Ta se označuje jako *NEMO*.

Centrální směrovač 6509 tyto funkce podporuje a proto předpokládám nasazení mobility v naší síti po určitém zajetí provozu na IPv6. Tuto vlastnost budou moci využít zaměstnanci jak při přechodu mezi sítěmi v rámci univerzity, tak především na svých cestách i doma.

8. Závěr

Při psaní své bakalářské práce jsem si rozšířil obzor týkající se jak standardů, tak i praktického nasazení protokolu IPv6. Při implementaci IPv6 jsem musel vycházet ze situace v síti, kdy jsou páteřní aktivní prvky výhradně zařízení firmy Cisco. Byť jsem zachoval při změně hlavního IPv6 směrovače na Cisco Catalyst 6509 statické směrování, musím říct že podpora dynamických směrovacích protokolů je výborná. Proto bych směrování IPv6 sítí těmito zařízeními bez výhrad doporučil. Poněkud horší byla situace na těchto prvcích s podporou DHCPv6, konkrétně chybějící funkce zprostředkovatele (relay). Při snaze o nasazení stavového přidělování adres to tak dlouhou dobu vypadalo na kompromis, kdy by na částech sítě zůstala automatická konfigurace adresy. Nicméně s posledními verzemi Cisco IOS se situace napravila a DHCPv6 se

tak využívá na všech uživatelských podsítích. DHCP server od ISC funguje dobře, pouze bych vytknul potřebu definovat subnet i pro síť připojenou k rozhraní serveru. Celkově tedy můžu kombinaci ISC DHCP serveru a Cisco směrovačů doporučit k realizaci stavového přidělování adres i na jiných sítích. Využití ISC BINDu jakožto DNS serveru i pro překlad IPv6 adres je bezproblémové a o jiné variantě serveru bych ani neuvažoval. Při zabezpečení provozu IPv6 sítě jsem narazil na problém při nastavování hraničního firewallu Cisco FWSM. Ten sice IPv6 podporuje, ne však jeho management konzole ASDM. Správce tak může firewall administrovat jen z příkazové řádky, což může být v závislosti na rozsahu přístupových pravidel docela časově náročné. Pokud tedy o tomto firewallu někdo uvažuje, měl by si pečlivě ověřit, co vše aktuální verze jeho OS a ASDM podporuje.

Jednou z hlavních motivací související s touto prací bylo rozšíření reálného poskytování služeb počítačové sítě i protokolem IPv6. To znamenalo nasazení IPv6 na produkčních serverech. Tento bod byl vzhledem k nedořešenému zabezpečení sítě splněn jen z části, kdy podporu IPv6 implementovali jen správci serverů s nekritickými aplikacemi. Nicméně i ostatní byli s problematikou seznámeni a očekávám v dohledné době nasazení na všech serverech. Výstupy této práce posloužily k vytvoření dokumentace o IPv6 na síti Ostravské univerzity dostupné na univerzitním portále i jako podklad pro proškolení počítačových techniků, starajících se o počítače jednotlivých fakult.

Seznam použité literatury

[1] SATRAPA, Pavel. *IPv6*. Praha : CZ.NIC, z.s.p.o., 2008. 357 s. Dostupné z WWW: <http://knihy.nic.cz/files/nic/edice/pavel_satrapa_ipv6_2008.pdf>. ISBN 978-80-904248-0-7

[2] POPOVICIU, Ciprian; LEVY-ABEGNOLI, Eric; GROSSETETE, Patrick. *Deploying IPv6 Networks*. [s.l.] : Cisco Press, 2006. 672 s. ISBN 1-58-705210-5

[3] Cisco Systems, Inc. *Cisco.com* [online]. 2009 [cit. 2010-05-06]. Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide, 4.0. Dostupné z WWW: <http://www.cisco.com/en/US/docs/security/fwsm/fwsm40/configuration/guide/ipv6_f.html>

[4] Cisco Systems, Inc. *Cisco.com* [online]. 2010 [cit. 2010-05-06]. Cisco IOS IPv6 Configuration Guide, Release 12.2SX. Dostupné z WWW: <http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/12_2sx/ipv6_12_2sx_book.html>

[5] Cisco Systems, Inc. *Cisco.com* [online]. 2010 [cit. 2010-05-06]. Cisco IOS IPv6 Command Reference. Dostupné z WWW: <http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html>

[6] WEISSMANN, Paul. *IPv6 Intelligence* [online]. 2007, 2010 [cit. 2010-05-06]. Dostupné z WWW: <http://www.ipv6int.net>

[7] LUDVIG, Michal. *Logix.cz* [online]. 2003 [cit. 2010-05-06]. IPv6 krok za krokem. Dostupné z WWW: <http://www.logix.cz/michal/doc/article.xp/ipv6-1>

[8] CESNET, z. s. p. o. *IPv6* [online]. 2008, 2010 [cit. 2010-05-06]. Dostupné z WWW: <http://www.ipv6.cz>

[9] CESNET, z. s. p. o. *Cesnet.cz* [online]. 1999, 2010 [cit. 2010-05-06]. IP verze 6. Dostupné z WWW: <http://www.cesnet.cz/ipv6>

[10] Internet Systems Consortium, Inc. *Isc.org* [online]. 2001, 2010 [cit. 2010-05-06]. DHCP. Dostupné z WWW: <http://www.isc.org/software/dhcp>

[11] Internet Systems Consortium, Inc. *Isc.org* [online]. 2001, 2010 [cit. 2010-05-06]. BIND. Dostupné z WWW: <http://www.isc.org/software/bind>

[12] IPv6 In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 3.11. 2005, 27.4. 2010 [cit. 2010-05-06]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/IPv6>

[13] Microsoft Corporation. *Microsoft.com* [online]. 2010 [cit. 2010-05-06]. Netsh Technical Reference. Dostupné z WWW: <http://technet.microsoft.com/en-us/library/cc725935%28WS.10%29.aspx>

[14] MRUGALSKI, Tomasz. *DHCPv6: Dibbler - a portable DHCPv6* [online]. 2003 [cit. 2010-05-06]. Dostupné z WWW: <http://klub.com.pl/dhcpv6/>

[15] *Linuxtopia.org* [online]. 2005 [cit. 2010-05-06]. Linux IPv6 HOWTO. Dostupné z WWW: <http://www.linuxtopia.org/online_books/network_administration_guides/Linux+IPv6-HOWTO/proc-sys-net-ipv6..html>